Artemis, AOD, and Analysis Flow

Kyle Cranmer
University of Wisconsin-Madison

Feb. 11, 2004
Analysis Tools

Outline:

- Overview of Artemis
- How I use Artemis for Physics Analysis
- How do Artemis and the AOD relate
- Handling Multiple Particle Definitions
- Proposals
- Comments collected on AOD

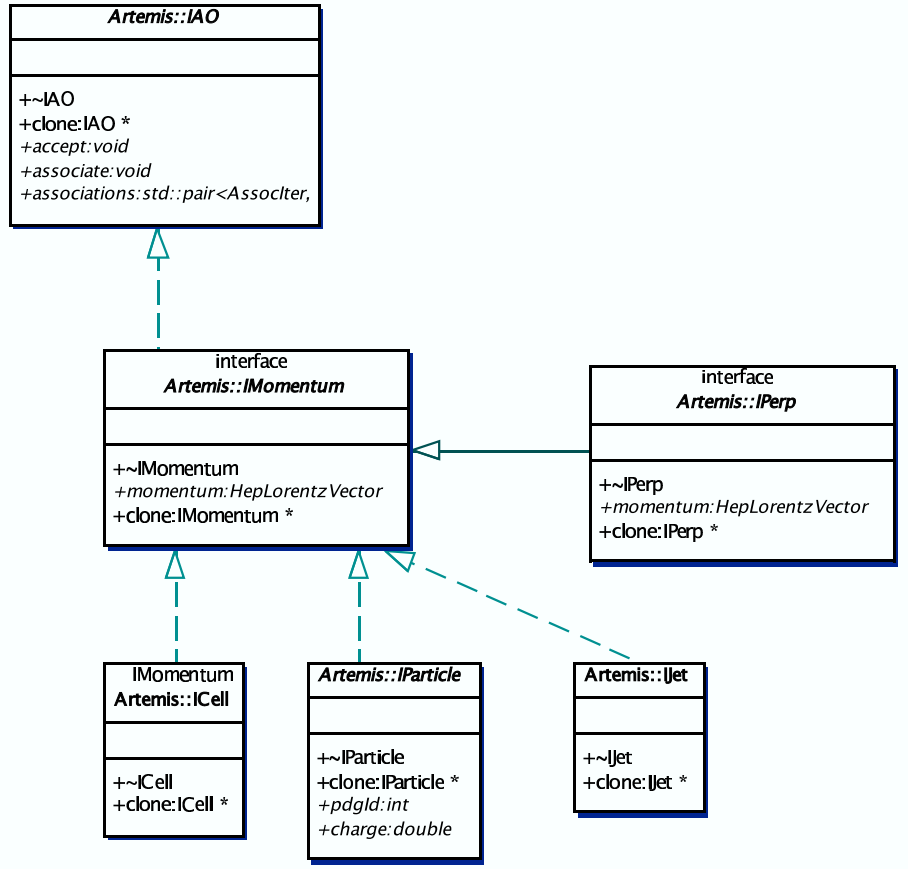
Overview

Artemis is an analysis framework to aid writing analyses & automate redundant tasks in athena.

Artemis provides:

- a standard interface to heterogeneous reconstruction classes (the design pattern *Adaptor*)
- fromTDS class to automate redundant tasks
- some common analysis untils. (like sort by $p_\perp$)

Users interact with high-level particles with momentum and Pid.



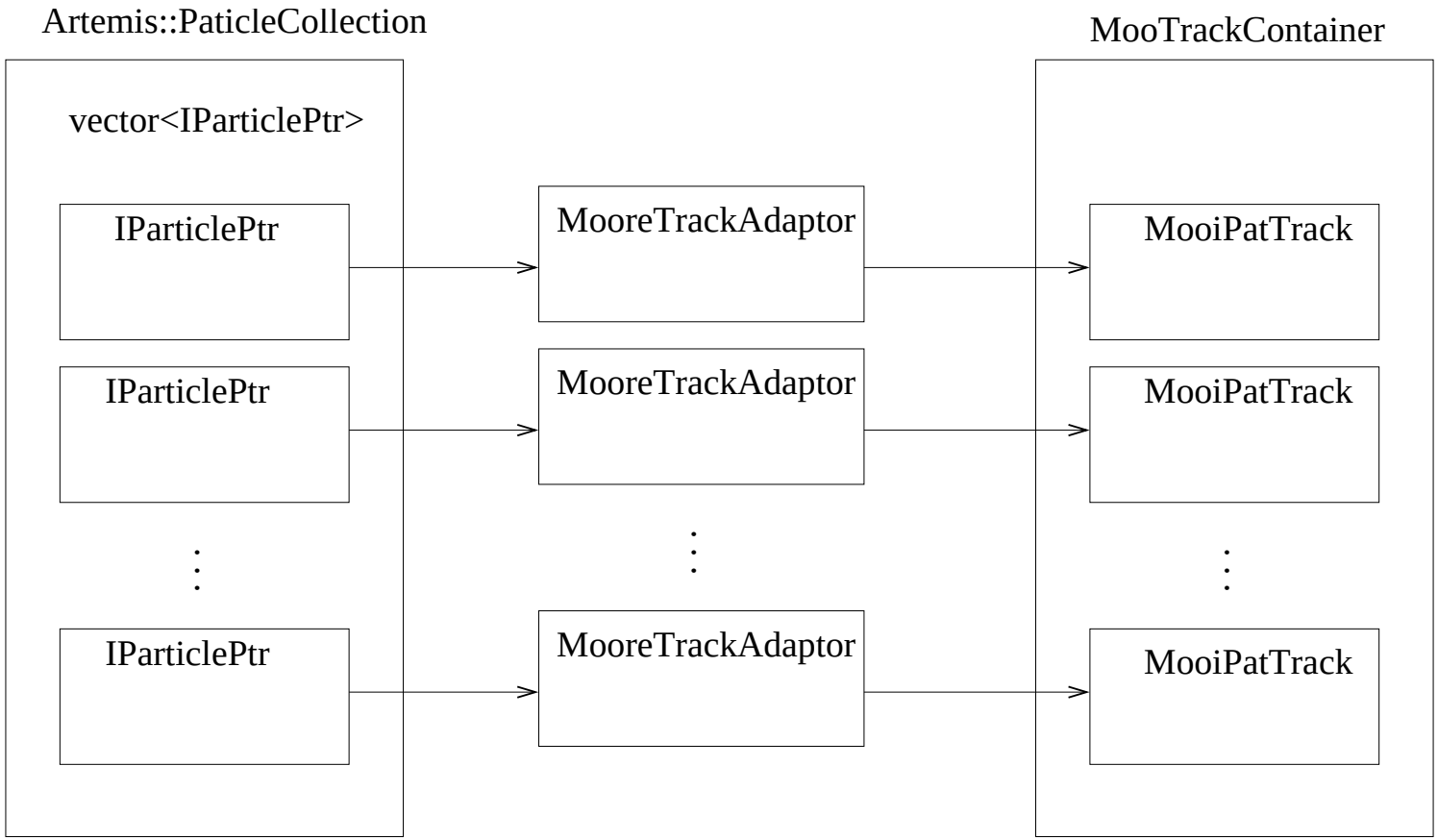
The `IAO` provides mechanics for visitor pattern. (Note: different use of `accept()` method)

The Interfaces are deliberately simple and will change as needed

Artemis provides Adaptors for the various reconstruction classes so that the user can use a common Artemis Interface.

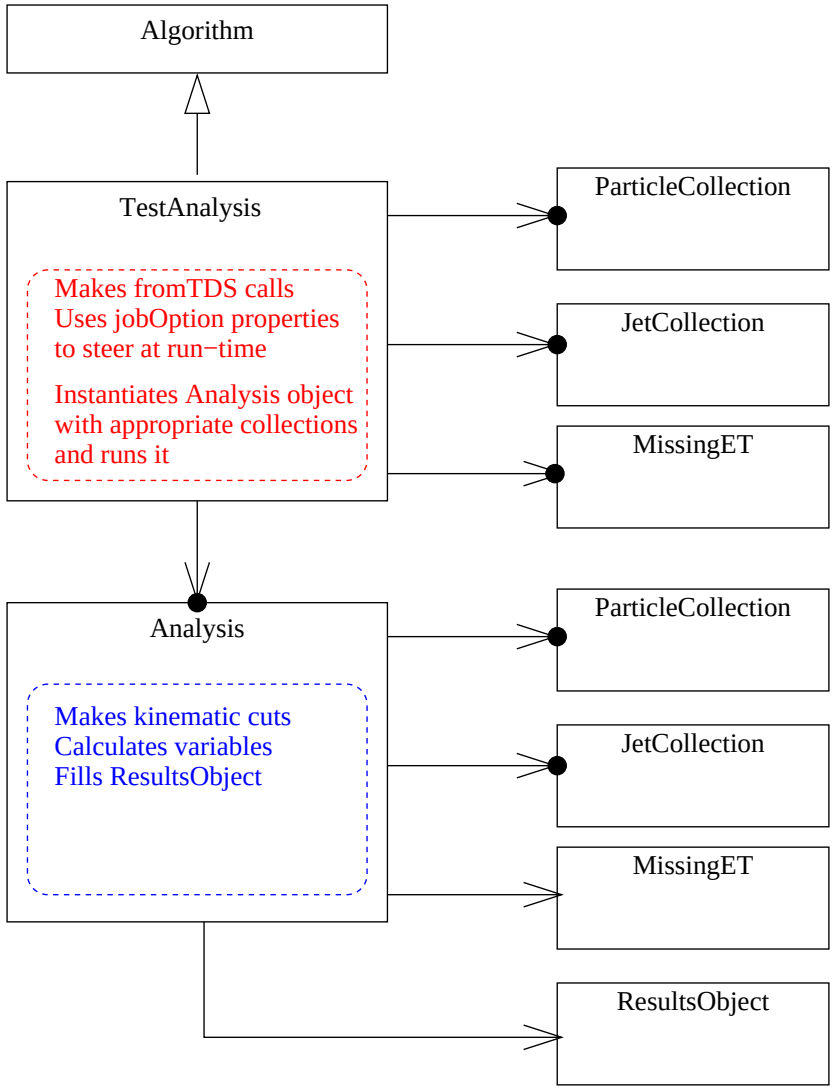
This makes it possible to re-use analysis code for different reconstruction classes.

fromTDS and Adaptor Example




```
ParticleCollection* myMooreMuons =          myFromTDS->get< MooreTrackCollection>()
ParticleCollection* myMuIdStandAloneMuons = myFromTDS->get< MuidTrackCollection>(m_muidStandAlonekey)
ParticleCollection* myMuIdComboMuons =      myFromTDS->get< MuidTrackCollection>(m_muidComboKey )
ParticleCollection* myMuonboxMuons =        myFromTDS->get< MuonboxTrackCollection>()
```

How I use Artemis



I develop my user-specific code within ArtemisUser

TestAnalysis is a class which inherits from a GaudiKernel Algorithm

TestAnalysis::execute() retrieves Collections from TDS & instantiates several Analysis Objects with their corresponding Collections

The Analysis object is *not* a Gaudi Algorithm

The Analysis object code can be reused for each permutation of the input Collections etc...

Multiple Analysis Streams make for easy event-by-event comparisons

Artemis uses Shared Pointers which are nice for memory management

Artemis has "Simple" objects which can be created by the user (not Adaptors).

If I want more detailed info than Artemis Interface, I can't get it easily.
$\Rightarrow$ Peter and I plan on providing pointer to raw TDS object (Adaptee)

Results object needs some work. Challenge is aggregation of different Analysis objects' results in TestAnalysis.

Artemis, AOD, and Analysis Flow

First reactions & misconceptions about the Artemis approach are:

- It's too simplistic
- It's too restrictive

Really the analysis is just factorized. Example: $H \longrightarrow 4e$

- Step 1: Decide what is an "electron" and how to calculate $\vec{p}$
- Step 2: Do Kinematic Analysis on collection of electrons

Most of the detailed reconstruction information
is needed only in Step 1.

I will talk about Step 1 in relation to Artemis, AOD,
and ATLAS Analysis in general.

How Artemis handles multiple particle definitions

Clearly, different analyses will have different definitions of what is a "good electron" or need to tune τ-effeciency w.r.t. jet-rejection, etc...

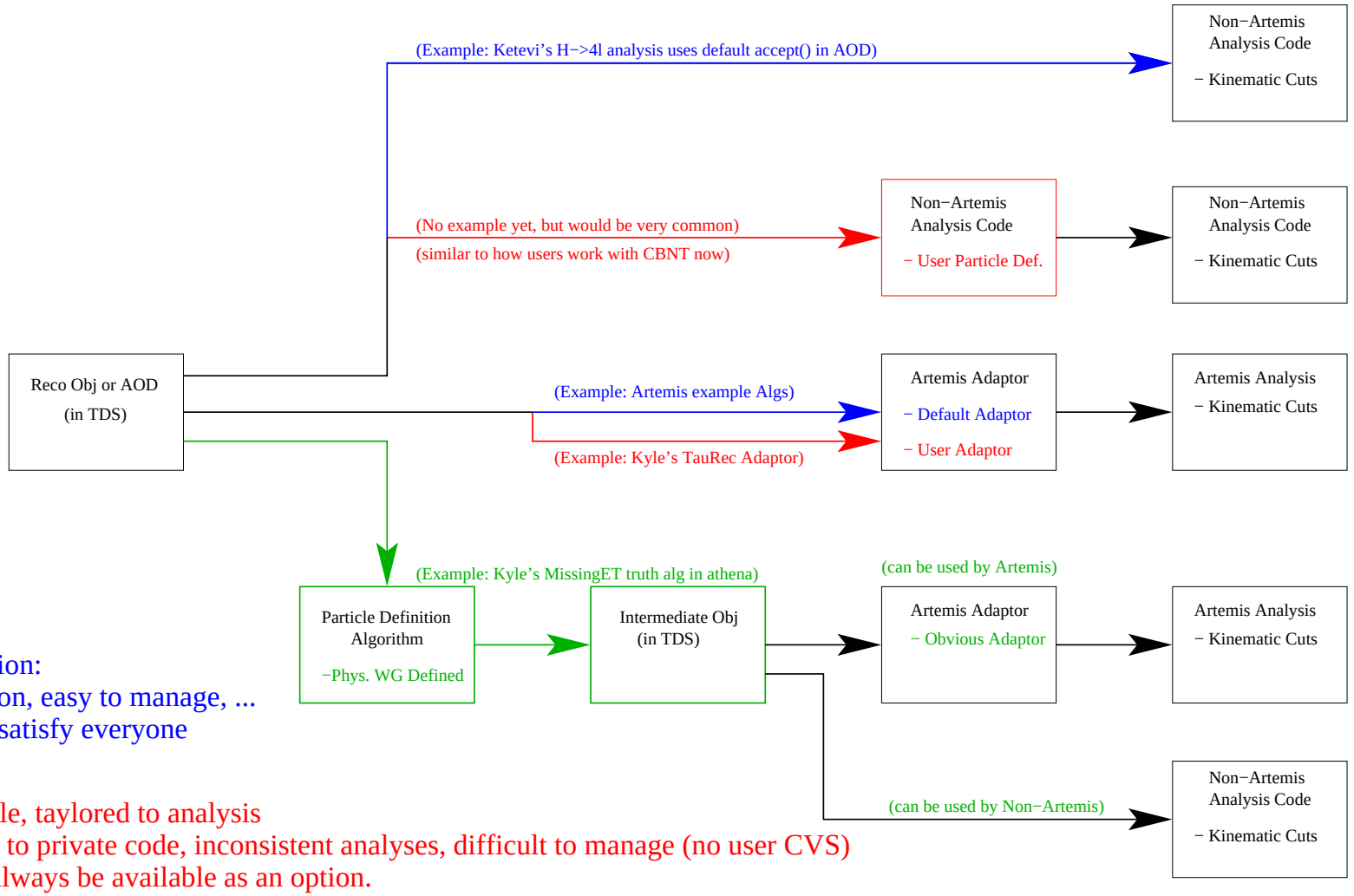
Step 1 is usally done in the Adaptor. For the most part there are only the "default" Adaptors.

How to add a new particle definition:

- Traditional way is to make a new Adaptor. Example: I wrote a TauRec τ Adaptor and a MuonBox μ Adaptor. I also wrote an alternate Egamma electron Adaptor to peform an Energy calibration.

- Alternatively, we can write a new algorithm that performs more sophisticated tasks and stores result in TDS. Then Artemis adapts the resulting object. Example: I modified MissingET package so Truth is stored in TDS and the MissingET Adaptor is unchanged.

Handling Multiple Particle Definitions



Default Definition:
- Pros − common, easy to manage, ...
- Cons − can't satisfy everyone

User Defined:
- Pros − flexible, taylored to analysis
- Cons − leads to private code, inconsistent analyses, difficult to manage (no user CVS)
- Note − will always be available as an option.

Phys WG Defined:
- Pros − tailored to analysis, easy to manage (in CVS Physics/ area), starting point for user−defined
- Cons − physics groups will need to manage this, need appropriate class def (AOD may be fine)
- Note − Surely the final goal is to have "official analysis" inside CVS

Proposals and Feedback

Proposal: Physics Group-Defined Particle Definitions

Request Physics (sub)groups to define
most common particle definitions.
(same as algo's in Fig 24 of RTF Report ATL-SOFT-2003-010 ?)

This could just be pseudo-code until AOD is decided,
but needs real list of variables or physics objects.

Would provide very useful information/feedback to those defining the AOD

Would aid in validation efforts

Necessary for green path in previous slide

Is a concrete, physics-oriented task to increase
interaction between physics groups & software developers

Proposal: List of Use-Cases

Compile a list of common Use-Cases so that we can quickly asses a proposed ESD/AOD or Analysis framework.

In conversations with Giacomo and Markus, having some example situations proved very useful.

Examples:

- User wants to provide customized cuts for electron id
- User wants to consider jet under light-quark c-quark, b-quark hypothesis
- User wants refit tracks
- User wants to refit primary vertex and recalculate 4-momenta
- User wants to reach back to the ESD

Comments on ESD/AOD (Prototype & Goal)

Markus:

- would like to see more of the object model.
- would like to see more standard Atlas classes so users don't have to rewrite code for ESD/AOD
- What about 4-momentum after primary vertex is refit?
- hopes to have enough track information to refit

Giacomo:

- expects that users will not have "their favorite variable" in AOD and will want to go back to ESD
- hoping to find an effecient and practical way to perserve ESD access by average user
- perhaps streams for ESD?

Peter:

- Thinks smart pointers could be very useful for AOD pointing back to ESD
- Could provide Control and log access to ESD.
- Good experiences with FastShower

Markus & Giacomo:

- Agree Use-Cases would be useful for conversation.

Appendix

The fromTDS Sequence diagram

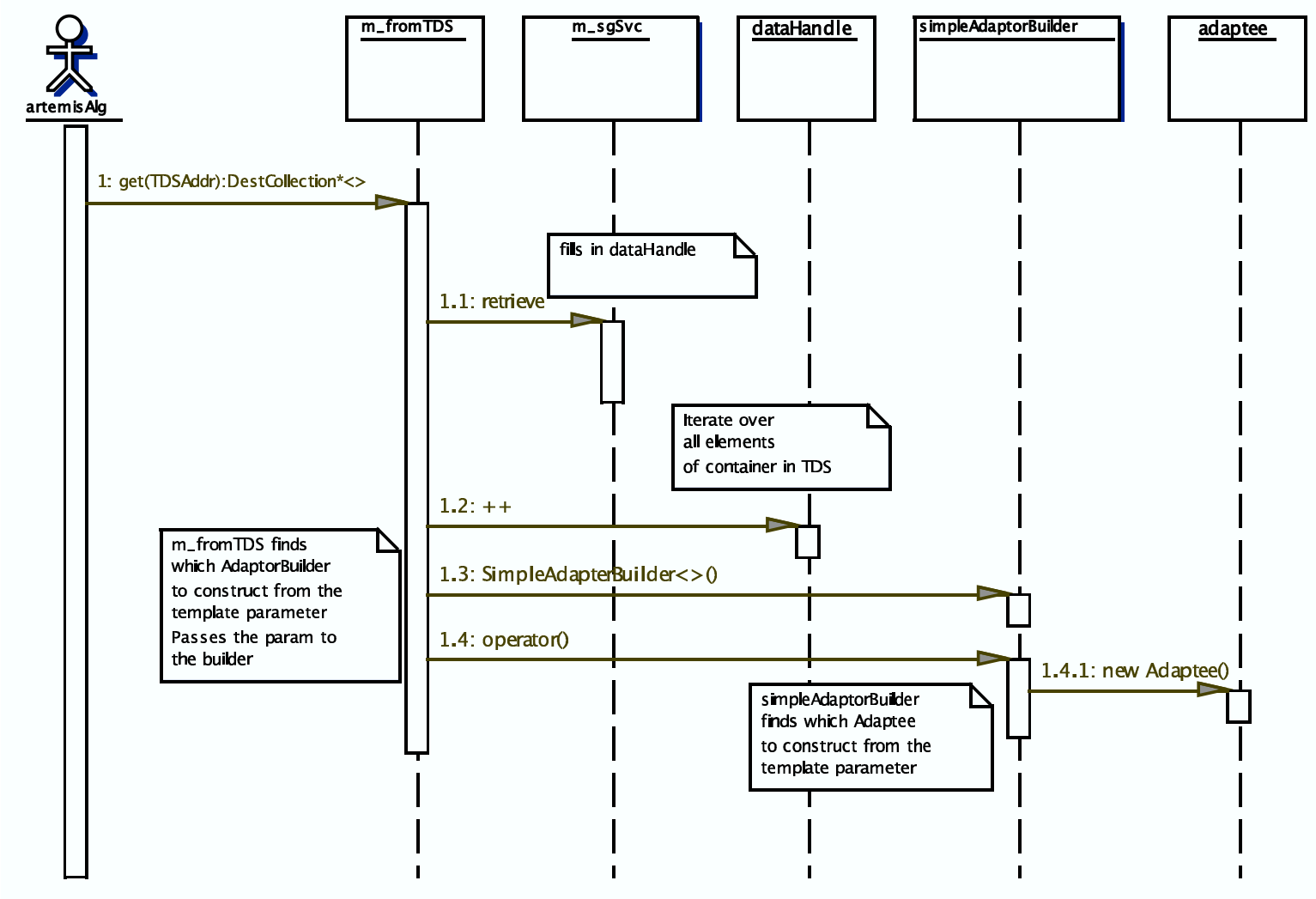


Figure 3: Sequence diagram for the FromTDS::get()

What does Artemis Adapt so far?

TDSclass	**Retrieves**
AtlfastJetCollection	Atlfast Jets
AtlfastBTauCollection	Atlfast Taus (tau tagged jets)
AtlfastRPCollection	Atlfast Reconstructed Particles (e, μ, γ)
AtlfastElectronCollection	Atlfast Reconstructed Particles (e only)
AtlfastPhotonCollection	Atlfast Reconstructed Particles (γ only)
AtlfastCellCollection	Atlfast Cells
RecoJetCollection	Reconstruction Jets
RecoCombinedJetCollection	Reconstruction Combined Jets
MooreTrackCollection	Moore μ-objects
MuidTrackCollection	Muid μ-objects
EgammaCollection	Reconstruction eGamma objects
ElectronCollection	Reconstruction eGamma objects (e only)
PhotonCollection	Reconstruction eGamma objects (γ only)
TightTauCollection	Reconstruction τ-objects
LooseTauCollection	Reconstruction τ-objects
TauCollection	Reconstruction τ-objects
AtlfastMissingET	Atlfast missing transverse energy
AtlfastEscapedET	Atlfast escaped (4-vec) missing transverse energy
ReconstructionMissingET	Reconstruction missing transverse energy

Table 2: Classes used to parameterise FromTDS::get() methods which retrieve information from the Transient Data Store and create Artemis Analysis objects, or collections of Artemis Analysis objects.